UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/665,932	09/18/2003	Chris Kirmse	08226/1203348-US2	6591

38880   7590   11/05/2009

Yahoo! Inc.
c/o DARBY & DARBY P.C.
P.O. BOX 770
Church Street Station
NEW YORK, NY 10008-0770

EXAMINER
LIM, SENG HENG

ART UNIT	PAPER NUMBER
3714	

MAIL DATE	DELIVERY MODE
11/05/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

<table>
<tr><td rowspan="2">Office Action Summary</td><td>Application No.
10/665,932</td><td>Applicant(s)
KIRMSE ET AL.</td><td></td></tr>
<tr><td>Examiner
SENG H. LIM</td><td>Art Unit
3714</td><td></td></tr>
</table>

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on *07 August 2009*.
2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) *1,8-10,17-20,23,24,28-33 and 35-105* is/are pending in the application.
   4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5) ☐ Claim(s) _____ is/are allowed.
6) ☒ Claim(s) *1,8-10,17-20,23,24,28-33 and 35-105* is/are rejected.
7) ☐ Claim(s) _____ is/are objected to.
8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.
10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All   b) ☐ Some *   c) ☐ None of:
      1. ☐ Certified copies of the priority documents have been received.
      2. ☐ Certified copies of the priority documents have been received in Application No. _____.
      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

A request for continued examination under 37 CFR 1.114, including the fee set

forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this

application is eligible for continued examination under 37 CFR 1.114, and the fee set

forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action

has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on

8/7/2009 has been entered.

Response to Amendment

This office action is in response to the amendment filed on 8/7/2009 in which

applicant amends claims 1, 17, 28, 33, 35, 38, 40, 52, 64, 76, 86, 96; and responds to

the claim rejections. Claims 1, 8-10, 17-20, 23-24, 28-33, 35-105 are pending.

Response to Arguments

Applicant's arguments filed 5/27/2009 that concern the amendments have been

answered in the rejection below.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
>
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
>
> the prior art are such that the subject matter as a whole would have been obvious at the time the
>
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
>
> Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 8-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Danieli et al (US 7240093 B1) in view of Beuk et al (US 5,774,673).

Re claims 1 & 8-10. Danieli et al discloses a game and messenger client server

system, comprising: a plurality of game clients (5:28-30); a game server (i.e. the player

hosting the game) including logic to operate a multiplayer game using inputs from and

outputs to an active game set of game clients including the plurality of game clients,

wherein game clients other than those in the active game set can join an active game

by supplying the game server with a reference to the active game (i.e. the game server

in this instance is the player hosting the game) (3:7-24); a plurality of messenger clients

and a messenger server (Fig. 6) including logic to forward messages from a sender

messenger client to a receiving messenger client; logic to couple a game client to a

messenger client to allow the game client to send the messenger client data used to

initiate joining a game, whereby a message sent by the messenger client includes the

data used to initiate joining a game; and logic to initiate a join of a game at an invitee

client, using data received in a message to the invitee (i.e. the host of the game sends out invitation to other players with a chat or messenger client to join a game, wherein the host initiates the start of a selected game after other players accept the invitation and join in) (9:58-62; 3:10-4:10; Fig. 19, 9). The system further comprising an icon that indicates a state of an inviter client, wherein the icon is a game-specific icon (7:32-40). The game and messenger client server system further comprises logic to generate a data file (i.e. a message) sent in response to a request from the invitee client (9:58-62).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being executable files 170 and the launch command 166, wherein these data would be sent to the messenger client of the host computer to the messenger client of the invitees to initiate the game).

Danieli et al noted that the player does not need to have the gaming utility opened or launched in order to receive an invitation to join a game. The player only needs to have the MSN messenger open (14:32-35). Once the invitation (which implicitly has a description of the game server because it needs information about the server or destination to connect to) to join is received by the invitee and when the invitee accepts, the gaming client is obviously connected to the game server since the game server is the player hosting the game. With that in mind, Danieli et al does not disclose that the data in the message or invitation sent by the messenger client comprises a command line executable for an invitee client to invoke a gaming client or utility. Beuk et al discloses wherein the data in a message or invitation sent by a

messenger client comprises a command line executable for an invitee client to execute

or invoke a gaming client or application (i.e. Beuk identifies the application to be run by

the received message and that appropriate application is executed/invoked based on

the identified application) (Abstract, 2:54-3:25, 9:21-28). Danieli et al and Beuk et al are

analogous art because they are from the same field of endeavor of using a messaging

client with a gaming client. At the time of invention a person of ordinary skill in the art

would have found it obvious to modify Danieli et al's system to incorporate Beuk's

method of invoking the gaming client with a start or joining message to connect to the

game server and would have been motivated to do so to provide alternative ways to

start a game.

Because the gaming and messenger client communicates with each other,

it is obvious or implicit that the game client determines that the messenger client

is able to receive messages. One of ordinary skilled in the art would have readily

recognizes the importance of the ability of two client communicating with one

another. The invention would not work if the gaming client determines the

messenger client is not able to receive messages.

Re Claims 17-20 & 23-24. Danieli et al discloses a method of operating a multi-

player game having a plurality of game clients and a plurality of messenger clients, the

plurality of game clients and plurality of messenger clients in communication with a

game server and a messenger server (Fig. 1), the method comprising: joining the game

by sending a reference to the game to the game server (i.e. the player hosting the

game); sending, from an inviter game client to an inviter messenger client, data (i.e.

chat invite) used to initiate joining the game and sending a message including the data

used to initiate joining the game to the messenger server; routing the message to an

invitee messenger client (i.e. sending the invitation to the invitee chat client); and using

the data in the routed message to invoke a game client and join the game (after the

invitee accepts the invitation, the game is launched such as "Age of Empires II" in

Figure 19) . The method includes sending, from the game server to the inviter game

client, a reference used to join the game and sends a message to a list of messenger

clients associated with the inviter messenger client, wherein an updated state (i.e. the

Status of the player) is perceptible by a user of the invitee messenger client; updating a

state of an icon (i.e. the icon next to player's name; Fig. 19) associated with the inviter

messenger client in response to receiving the message; and sending a request for a

game data file to the game server wherein the game data file includes a reference to the

game locally (i.e. all the invitee would obvious require a request to the game server for

the game data to play the game Age of Empires II for instance has to locally be installed

in the invitee client's computer to execute the game) (3:10-4:10).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to

the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being

executable files 170 and the launch command 166, wherein these data would be sent to

the messenger client of the host computer to the messenger client of the invitees to

initiate the game).

Danieli et al noted that the player does not need to have the gaming utility

opened or launched in order to receive an invitation to join a game. The player only

needs to have the MSN messenger open (14:32-35). Once the invitation (which

implicitly has a description of the game server because it needs information about the

server or destination to connect to) to join is received by the invitee and the invitee

accepts, the gaming client is obviously connected to the game server. With that in mind,

Danieli et al does not disclose wherein the data in the message or invitation sent by the

messenger client comprises a command line and registry entry executable for an invitee

client to invoke the gaming client or utility to connect to the game server since the game

server is the player hosting the game. Beuk et al discloses that the data in a message

or invitation sent by a messenger client comprises a command line and registry entry

executable for an invitee client to execute or invoke a gaming client or application (i.e.

Beuk identifies the application to be run by the received message and that appropriate

application is executed/invoked based on the identified application. Additionally, it's

obvious that Beuk et al has a registry entry for an invitee client because when a

program is installed a registry key is created) (Abstract, 2:54-3:25, 9:21-28). Danieli et al

and Beuk et al are analogous art because they are from the same field of endeavor of

using a messaging client with a gaming client. At the time of invention a person of

ordinary skill in the art would have found it obvious to modify Danieli et al's system to

incorporate Beuk's method of invoking the gaming client with a start or joining message

to connect to the game server and would have been motivated to do so to provide

alternative ways to start a game.

Because the gaming and messenger client communicates with each other,

it is obvious or implicit that the game client determines that the messenger client

is able to receive messages. One of ordinary skilled in the art would have readily recognizes the importance of the ability of two client communicating with one another. The invention would not work if the gaming client determines the messenger client is not able to receive messages.

Re claims 28-32. Danieli et al discloses a method of operating a multi-player game having an inviter client, an invitee client, and a server, the method comprising: invoking an inviter game client at the inviter client; connecting the inviter game client to the game by sending a reference to the game to the server; creating a message at the inviter client containing data used for invoking an invitee game client and for joining the game; routing the message to the invitee client; and using the data in the message to invoke the invitee game client and join the game (i.e. the server/host of the game sends out chat invitation to other players on his list and after invitees accept to join, the host executes or invoke the game to other players). The message is created at the inviter client/server and routes the message by using TCP/IP (2:6-10). The message is sent to a second server such as the messenger server (3:10-4:10).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being executable files 170 and the launch command 166, wherein these data would be sent to the messenger client of the host computer to the messenger client of the invitees to initiate the game).

Danieli et al noted that the player does not need to have the gaming utility opened or launched in order to receive an invitation to join a game. The player only

needs to have the MSN messenger open (14:32-35). Once the invitation (which

implicitly has a description of the game server because it needs information about the

server or destination to connect to) to join is received by the invitee and the invitee

accepts, the gaming client is obviously connected to the game server. With that in mind,

Danieli et al does not disclose wherein the data in the message or invitation sent by the

messenger client comprises a command line executable for an invitee client to invoke

the gaming client or utility to connect to the game server since the game server is the

player hosting the game. Beuk et al discloses that the data in a message or invitation

sent by a messenger client comprises a command line executable for an invitee client to

execute or invoke a gaming client or application (i.e. Beuk identifies the application to

be run by the received message and that appropriate application is executed/invoked

based on the identified application. Additionally, it's obvious Beuk et al has a registry

entry for an invitee client because when a program is installed a registry key is created)

(Abstract, 2:54-3:25, 9:21-28). Danieli et al and Beuk et al are analogous art because

they are from the same field of endeavor of using a messaging client with a gaming

client. At the time of invention a person of ordinary skill in the art would have found it

obvious to modify Danieli et al's method to incorporate Beuk's method of invoking the

gaming client with a start or joining message to connect to the game server and would

have been motivated to do so to provide alternative ways to start a game.

Because the gaming and messenger client communicates with each other,

it is obvious or implicit that the game client determines that the messenger client

is able to receive messages. One of ordinary skilled in the art would have readily

recognizes the importance of the ability of two client communicating with one another. The invention would not work if the gaming client determines the messenger client is not able to receive messages.

Re claim 33. Danieli et al discloses a game and messenger client server system, comprising: a plurality of game clients including an inviter and an invitee game client; a plurality of messenger clients including an inviter and invitee messenger client (i.e. chat client); a server including logic to operate a multiplayer game using inputs from and outputs to an active game set of game clients of the plurality of game clients, wherein game clients other than those in the active game set can join an active game by supplying the server with a reference to the active game (i.e. such as an IP address) (3:10-13, 10:43-48); logic to couple the inviter game client to the inviter messenger client to allow the inviter game client to send the inviter messenger client data used to initiate joining a game, whereby a message sent by the inviter messenger client includes the data used to initiate joining a game; and logic to initiate a join of a game at the invitee game client, using data received in a message to the invitee messenger client, wherein the inviter messenger client includes logic to forward messages to the invitee messenger client (i.e. the server/host of the game sends out chat invitation to other players on his list and after invitees accept to join, the host executes or invoke the game to other players). (3:25-53).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being executable files 170 and the launch command 166, wherein these data would be sent to

the messenger client of the host computer to the messenger client of the invitees to

initiate the game).

Danieli et al noted that the player does not need to have the gaming utility

opened or launched in order to receive an invitation to join a game. The player only

needs to have the MSN messenger open (14:32-35). Once the invitation (which

implicitly has a description of the game server because it needs information about the

server or destination to connect to) to join is received by the invitee and the invitee

accepts, the gaming client is obviously connected to the game server. With that in mind,

Danieli et al does not disclose wherein the data in the message or invitation sent by the

messenger client comprises a command line executable for an invitee client to invoke

the gaming client or utility to connect to the game server. Beuk et al discloses that the

data in a message or invitation sent by a messenger client comprises a command line

executable for an invitee client to execute or invoke a gaming client or application (i.e.

Beuk identifies the application to be run by the received message and that appropriate

application is executed/invoked based on the identified application. Additionally, it's

obvious Beuk et al has a registry entry for an invitee client because when a program is

installed a registry key is created) (Abstract, 2:54-3:25, 9:21-28). Danieli et al and Beuk

et al are analogous art because they are from the same field of endeavor of using a

messaging client with a gaming client. At the time of invention a person of ordinary skill

in the art would have found it obvious to modify Danieli et al's system to incorporate

Beuk's method of invoking the gaming client with a start or joining message to connect

to the game server and would have been motivated to do so to provide alternative ways to start a game.

Because the gaming and messenger client communicates with each other, it is obvious or implicit that the game client determines that the messenger client is able to receive messages. One of ordinary skilled in the art would have readily recognizes the importance of the ability of two client communicating with one another. The invention would not work if the gaming client determines the messenger client is not able to receive messages.

Re claims 35-39. Danieli et al discloses a program and method for providing a multi user networked computing environment, the method using an activity server and a messenger server, where the activity server and the messenger server are configured to communicate with a plurality of user computer systems, the user computer system including an activity client where the user computer system executes a user interface operated by a human user and is further configured to engage an activity using the activity client, wherein the user interface includes a display device and a user input device, wherein the user computer system is coupled to a network for exchanging information with the activity server and the messenger server (Fig. 1, 6), the method comprising: accepting signals from the user input device to engage the activity or game using the activity or game client (i.e. creating an invitation); presenting one or more preferences to the user computer system, where the one or more preferences are associated with activities or games (i.e. player's on the inviter chat client); selecting at least one preference to join the activity or game; invoking the selected activity with a

messenger client; providing to the messenger server a user state and a reference to the

activity or game in which the user is participating; and presenting to another user

associated with at least one of the plurality of user computer systems the user state and

the reference to the activity or game (i.e. the server/host of the game sends out chat

invitation to other players on his list and after invitees accept to join, the host executes

or invoke the game to other players), (3:10-53).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to

the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being

executable files 170 and the launch command 166, wherein these data would be sent to

the messenger client of the host computer to the messenger client of the invitees to

initiate the game).

Danieli et al noted that the player does not need to have the gaming utility

opened or launched in order to receive an invitation to join a game. The player only

needs to have the MSN messenger open (14:32-35). Once the invitation (which

implicitly has a description of the game server because it needs information about the

server or destination to connect to) to join is received by the invitee and the invitee

accepts, the gaming client is obviously connected to the game server. With that in mind,

Danieli et al does not disclose wherein the data in the message or invitation sent by the

messenger client comprises a command line executable for an invitee client to invoke

the gaming client or utility to connect to the game server. Beuk et al discloses that the

data in a message or invitation sent by a messenger client comprises a command line

executable for an invitee client to execute or invoke a gaming client or application (i.e.

Beuk identifies the application to be run by the received message and that appropriate

application is executed/invoked based on the identified application. Additionally, it's

obvious Beuk et al has a registry entry for an invitee client because when a program is

installed a registry key is created) (Abstract, 2:54-3:25, 9:21-28). Danieli et al and Beuk

et al are analogous art because they are from the same field of endeavor of using a

messaging client with a gaming client. At the time of invention a person of ordinary skill

in the art would have found it obvious to modify Danieli et al's system to incorporate

Beuk's method of invoking the gaming client with a start or joining message to connect

to the game server and would have been motivated to do so to provide alternative ways

to start a game.

Because the gaming and messenger client communicates with each other,

it is obvious or implicit that the game client determines that the messenger client

is able to receive messages. One of ordinary skilled in the art would have readily

recognizes the importance of the ability of two client communicating with one

another. The invention would not work if the gaming client determines the

messenger client is not able to receive messages.

Re claims 40-51 & 96-105. Danieli et al discloses a logic and computer program

product for use at an invitee client to initiate joining by an invitee game client to an

active game that is hosted by a game server and to which an inviter game client is

joined, the invitee client including an invitee messenger client for receiving in at least

one message from an inviter messenger client data used to initiate joining a game, the

logic comprising: invocation logic for using the data to invoke the invitee game client

and connect the invitee game client to the game server, wherein the data includes a reference to the game server and a reference to the active game, the inviter and invitee game clients being respectively associated with the inviter and invitee messenger clients. The data used to initiate joining a game includes a game server network address that identifies the game server, a game identifier that identifies the active game on the identified game server, and a port identifier that identifies a port on the identified game server (3:10-13, 10:43-48). Danieli also discloses the logic for activating the invocation logic in response to action by a user (10:14-17); for displaying a buddy list of the invitee messenger client and an indication that the invitee game client may join an active game which a member of the buddy list is playing (Fig. 8); for displaying a game-specific icon identifying the active game (Fig. 19); for use at an invitee client wherein the invitee messenger client is associated with a member of a buddy list of the inviter messenger client (Fig. 18); for use at an invitee client wherein the invitee messenger and game clients reside at a first computer system, and the inviter messenger and game clients reside at a second computer system (Fig. 1, 8, 14); for sending to other messenger clients at least one message including a reference to an active game (3:10-13, 45-50); for use at an invitee client wherein the invitee messenger client is operable to receive the at least one message inherently via a messenger server and to read at least one registry entry usable to invoke the invitee game client; for use at an invitee client wherein the invitee messenger client is operable to receive at least one message including a reference to a potential game (3:10-13, 45-50).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being executable files 170 and the launch command 166, wherein these data would be sent to the messenger client of the host computer to the messenger client of the invitees to initiate the game).

Danieli et al noted that the player does not need to have the gaming utility opened or launched in order to receive an invitation to join a game. The player only needs to have the MSN messenger open (14:32-35). Once the invitation (which implicitly has a description of the game server because it needs information about the server or destination to connect to) to join is received by the invitee and the invitee accepts, the gaming client is obviously connected to the game server. With that in mind, Danieli et al does not disclose wherein the data in the message or invitation sent by the messenger client comprises a command line executable for an invitee client to invoke the gaming client or utility to connect to the game server. Beuk et al discloses wherein the data in a message or invitation sent by a messenger client comprises a command line executable for an invitee client to execute or invoke a gaming client or application (i.e. Beuk identifies the application to be run by the received message and that appropriate application is executed/invoked based on the identified application.) (Abstract, 2:54-3:25, 9:21-28). Danieli et al and Beuk et al are analogous art because they are from the same field of endeavor of using a messaging client with a gaming client. At the time of invention a person of ordinary skill in the art would have found it obvious to modify Danieli et al's logic to incorporate Beuk's logic of invoking the gaming

client with a start or joining message to connect to the game server and would have

been motivated to do so to provide alternative ways to start a game.

Because the gaming and messenger client communicates with each other,

it is obvious or implicit that the game client determines that the messenger client

is able to receive messages. One of ordinary skilled in the art would have readily

recognizes the importance of the ability of two client communicating with one

another. The invention would not work if the gaming client determines the

messenger client is not able to receive messages.

Re claims 52-95. Danieli et al discloses a logic with computer program product

comprising program code and method of operating an invitee client to initiate joining by

an invitee game client to an active game that is hosted by a game server and to which

an inviter game client is joined, the invitee client including an invitee messenger client

for receiving in at least one message from an inviter messenger client data used to

initiate joining a game, the method comprising: invoking the invitee game client using

the data; and connecting the invitee game client to the game server using the data,

wherein the data includes a reference or identifier such as an IP address to the game

server and a reference to the active game, the inviter and invitee game clients being

respectively associated with the inviter and invitee messenger clients. User initiates

joining to the active game in response to action by a user (3:10-13, 45-50, 10:43-48).

The method further comprising displaying a buddy list of the invitee messenger client

and an indication that the invitee game client may join an active game which a member

of the buddy list is playing (Fig. 8). The method further comprising displaying a game-

specific icon identifying the active game (Fig. 19). The invitee messenger client is associated with a member of a buddy list of the inviter messenger client (Fig. 18). The invitee messenger and game clients reside at a first computer system, and the inviter messenger and game clients reside at a second computer system (Fig. 1). The method further comprising sending to other messenger clients at least one message including a reference to an active game (3:10-13, 45-50, 10:43-48).

Danieli et al disclose the game client (72 & 162: Fig. 19) to create and send to the messenger client (32: Fig. 19) data used to initiate joining a game (i.e. data being executable files 170 and the launch command 166, wherein these data would be sent to the messenger client of the host computer to the messenger client of the invitees to initiate the game).

Danieli et al noted that the player does not need to have the gaming utility opened or launched in order to receive an invitation to join a game. The player only needs to have the MSN messenger open (14:32-35). Once the invitation (which implicitly has a description of the game server because it needs information about the server or destination to connect to) to join is received by the invitee and the invitee accepts, the gaming client is obviously connected to the game server. With that in mind, Danieli et al does not disclose wherein the data in the message or invitation sent by the messenger client comprises a command line executable for an invitee client to invoke the gaming client or utility to connect to the game server. Beuk et al discloses wherein the data in a message or invitation sent by a messenger client comprises a command line executable for an invitee client to execute or invoke a gaming client or application

(i.e. Beuk identifies the application to be run by the received message and that

appropriate application is executed/invoked based on the identified application.)

(Abstract, 2:54-3:25, 9:21-28). Danieli et al and Beuk et al are analogous art because

they are from the same field of endeavor of using a messaging client with a gaming

client. At the time of invention a person of ordinary skill in the art would have found it

obvious to modify Danieli et al's logic to incorporate Beuk's logic of invoking the gaming

client with a start or joining message to connect to the game server and would have

been motivated to do so to provide alternative ways to start a game.

Danieli does not expressly disclose validating the potential game as legitimate.

Beuk et al discloses validating the potential game as legitimate by verifying with the

activation unit (Abstract). At the time of invention a person of ordinary skill in the art

would have found it obvious to incorporate the verification of the potential game as

legitimate to make sure player's have legitimate games.

Because the gaming and messenger client communicates with each other,

it is obvious or implicit that the game client determines that the messenger client

is able to receive messages. One of ordinary skilled in the art would have readily

recognizes the importance of the ability of two client communicating with one

another. The invention would not work if the gaming client determines the

messenger client is not able to receive messages.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created

doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the

unjustified or improper timewise extension of the "right to exclude" granted by a patent

and to prevent possible harassment by multiple assignees.   A nonstatutory

obviousness-type double patenting rejection is appropriate where the conflicting claims

are not identical, but at least one examined application claim is not patentably distinct

from the reference claim(s) because the examined application claim is either anticipated

by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140

F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29

USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir.

1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422

F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163

USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d)

may be used to overcome an actual or provisional rejection based on a nonstatutory

double patenting ground provided the conflicting application or patent either is shown to

be commonly owned with this application, or claims an invention made as a result of

activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a

terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with

37 CFR 3.73(b).

Claims 1, 17, & 33 are rejected on the ground of nonstatutory obviousness-type

double patenting as being unpatentable over 1 & 13 of U.S. Patent No. 6699125 in view

of Danieli et al (US 7240093 B1). Although the conflicting claims are not identical, they

are not patentably distinct from each other because both claim a method of operating a

game and messenger client server system comprising of a plurality of game clients, a

game server, plurality of messenger clients, a messenger server, and logic to couple

game client to messenger client and initiate a join of a game at an invitee client. U.S.

Patent No. 6699125 discloses using data received in a message to the invitee to include

a reference (i.e. description of the game server) to a game server and commands

usable or executable to invoke a game client for an invitee client. U.S. Patent No.

6699125 is silent in the game client creating and sending to the messenger client data

used to initiate joining a game; however, Danieli et al disclose the game client (72 &

162: Fig. 19) to create and send to the messenger client (32: Fig. 19) data used to

initiate joining a game (i.e. data being executable files 170 and the launch command

166, wherein these data would be sent to the messenger client of the host computer to

the messenger client of the invitees to initiate the game). At the time of invention a

person of ordinary skill in the art would have found it obvious with the evidence by

Danieli et al that the game client can create and send data to the messenger client data

to initiate a game because the invitee's game client needs data and information of the

game in order to play the game. **Because the gaming and messenger client**

communicates with each other, it is obvious or implicit that the game client

determines that the messenger client is able to receive messages. One of

ordinary skilled in the art would have readily recognizes the importance of the

ability of two client communicating with one another. The invention would not

work if the gaming client determines the messenger client is not able to receive

messages.

Correspondence

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to SENG H. LIM whose telephone number is (571)270-

3301. The examiner can normally be reached on 9:30-6:00, Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Peter Vo can be reached on 571-272-4690. The fax phone number for the

organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/S. H. L./

Examiner, Art Unit 3714

/Corbett Coburn/
Primary Examiner
AU 3714